# QoS as Middleware: Bandwidth Brokering System Design*

***Gary Hoo and William Johnston, Lawrence Berkeley National Laboratory***
***Ian Foster and Alain Roy, Argonne National Laboratory and University of Chicago***

We describe an approach to providing reservable bandwidth as a service for distributed computing environments such as computational Grids that aggregate resources to solve a single problem, remote instruments that operate on a schedule and that depend on communications and other Grid resources in order to function, etc.

Our system provides the mechanisms for applications to make and use advance reservations for bandwidth. These mechanisms are built on top of the existing work in IP differentiated services. (The description in this paper is given in terms of IP differentiated service classes, but the intent is that it could also be applied to, e.g., a dynamic ATM circuit set up mechanism where QoS can be specified.)

This document provides an overview of the bandwidth brokering system's model and design. It also describes the details of the components of the architecture, explains the flow of control among the components and describes typical use.

The system performs reservation-based scheduling and supplies information to support rudimentary negotiation by the broker element. Our service works with current computational Grids by extending the Globus Toolkit. In particular, we use the Globus Resource Specification Language (RSL) to express reservation needs and the Globus GRAM interface to local resource managers for the service itself. Globus' DUROC will provide the initial broker service. The system uses the Akenti access control system to provide policy-based use of a class of service or a circuit-based allocation.

## 1.0 Overview

We describe our approach to providing quality of service in an IP-based (inter)network that supports some means of differentiating between classes of service. Our work will extend this infrastructure by providing a way to request and to confirm reservation of a service class end-to-end. Section 1.1 presents the system's target audience and its requirements. Section 1.2 describes how the required per-node resource reservation is performed using a slot scheduler, and Section 1.3 discusses access control for the reservation mechanism. Section 1.4 explains how a broker performs advanced, end-to-end bandwidth reservation using multiple slot schedulers, with Section 1.5 exploring the required access-control mechanism. Section 1.6 describes resource claiming.

### 1.1 Requirements of users

We are addressing QoS tailored to serve a rather specialized audience. One type of user will be scientists who are performing experiments on singular instruments such as the LBL Advanced

*This work is supported by the U. S. Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office (http://www.er.doe.gov/production/octr/mics), under contract DE-AC03-76SF00098 with the University of California and contract XXX with the University of Chicago. The authors may be contacted at:hoo@george.lbl.gov, wejohnston@lbl.gov, itf@mcs.anl.gov, roy@mcs.anl.gov. This document is report LBNL-42947.

Light Source (ALS). Because such instruments cannot be replicated at each potential user's site, access requires either physical presence or a robust, high-speed computing framework that allows the user to control the instrument remotely, collect the data from experiments, and possibly process and analyze the data in real time. Remote access thus will require concurrent use of large amounts of computing, storage, and networking resources. Moreover, use of these resources must be scheduled in advance because the scientific instruments are so scheduled.

The same restriction holds true for the other type of targeted user, who needs to harness heterogeneous distributed resources to perform large-scale computation. These resources—again, computational, storage, and network—must also be scheduled in advance to ensure that they are all simultaneously available.

### 1.2 Per-node resource reservation

Our model allocates time in IP differentiated service classes [2]. Each class is associated with an elevated queueing priority service class. The elevated priority classes have upper bounds, set by the network service provider (NSP), on the total bandwidth allocation. Each node (host) from, to, and through which network traffic will be sent must set aside physical resources, such as buffer space, for the traffic associated with each service class.

The reservation unit is a *slot* - i.e., a period of time with a defined beginning and end, and an associated bandwidth. Each reservation decrements the bandwidth available in the class over a given time interval. The sum of the bandwidths in all of the slot allocations never exceeds the maximum bandwidth defined for the service class. The result is that the classes are never oversubscribed and bandwidth is effectively reserved.

The entity that allocates slots on behalf of a resource, such as a service class supported by a router, is the *slot scheduler.* The slot scheduler is controlled by a *resource manager* that communicates with external entities on the scheduler's behalf. One of the resource manager's main responsibilities is to ensure the slot scheduler's integrity and security by performing authentication and authorization checks on all reservation and claiming requests before allowing the slot scheduler to act on them.

### 1.3 Securing per-node resource reservation

When a resource manager receives a reservation request, its first action is to determine, then to verify, the requestor's identity. Because the resource manager will operate within the Globus environment [5], it will use the Globus authentication mechanism [7] to carry out that task.

All those who hold administrative (e.g., veto) authority over a resource, collectively known as the resource's *stakeholders*, will want to impose restrictions on its use. Such restrictions, which constitute the *policy* governing access to the resource, could take the form of time-of-day use limits, group membership, or application traffic type, among other things, in addition to straightforward identity-based access control. Following authentication of the requestor, therefore, the resource manager must determine the elements of the access-control policy, and check whether the requestor has satisfied them. Resource managers that are responsible for initial request admission, that is, those that first receive a broker's resource reservation request, will use Akenti [9] to perform such access-control checks. (Resource managers representing "interior nodes" in the network need not and probably will not redo this policy check; see Section 1.5.)

A successful reservation is represented by a cryptographic token (a digitally signed document) returned by the resource manager to the requestor. The token is the reservation guarantee. The token can be passed to another party, such as another resource manager (see Section 1.5), and the recipient, by verifying the signature, can be sure that the broker has not tampered with the token.

The resource manager performing initial admission control must have access to a trusted identity certifier, such as a certificate authority, to perform Akenti policy-checking. This implies a certain level of trust between the resource manager and the requestor's domain.

### 1.4 End-to-end bandwidth reservation

A client that wishes to communicate with guaranteed bandwidth uses a broker service to secure simultaneous reservations with all required resources. In the case of a network path, the resources are the premium service classes provided by the NSP. Due to restrictions in its network and/or at the egress points, the NSP probably cannot support simultaneous maximum use by all customers. Therefore, all reservations, even those made by otherwise independent entities—e.g. in Figure 1, C1 to C2—must account for prior or total use of the limited resource in the network interior, e.g., the network elements represented by the NSP1-1, 1-2, and 1-3 reservation managers.

An operational description of the model follows, and is illustrated in Figure 1. Client C1 asks the broker for premium bandwidth to C2 (step 1). The broker contacts the *first-hop resource manager*—the manager associated with the service provider ingress gateway closest to the client—and requests the reservation (2). The resource manager authenticates C1 and checks its access privileges for the premium bandwidth (2a). It responds with two types of information: its own, local slot availability (and/or related slots), and the identity of the next resource manager that must be contacted (2b). In this way, the broker is guided through the network to each resource that must be reserved (3, 4, 5, 6). (Steps 6a and 6b, involving access control at C2, are described in Section 1.5.)

Each resource manager (labeled "NSPn-m" in the figure), like the first-hop manager, responds with slot availability and next-hop location. The same hop-by-hop negotiation process is also performed at the borders between NSPs, with the manager at the egress node of the upstream NSP being responsible for directing the broker to the ingress resource manager for the downstream NSP.

The simplest negotiation for a given bandwidth during a given time period is to obtain one or more contiguous slots, comprising a *slot window*, with the response of all the managers being positive or negative. More sophisticated negotiation would require the managers to respond with a list of slot windows of the requested duration within some range of the original starting time. With this information the broker could pick, perhaps based on some client criteria, some window when all resources are available.

### 1.5 Securing end-to-end bandwidth reservation

To a first-hop resource manager, the requestor is the combination of application and end user invoking the broker. However, other resource managers generally do not have the kind of trust relationship with the end user's trust domain that permits the kind of access control verification described in Section 1.3.

Instead, resource managers enjoy bilateral trust relationships with one another. A given resource manager is configured to know its upstream and downstream neighbor managers and will

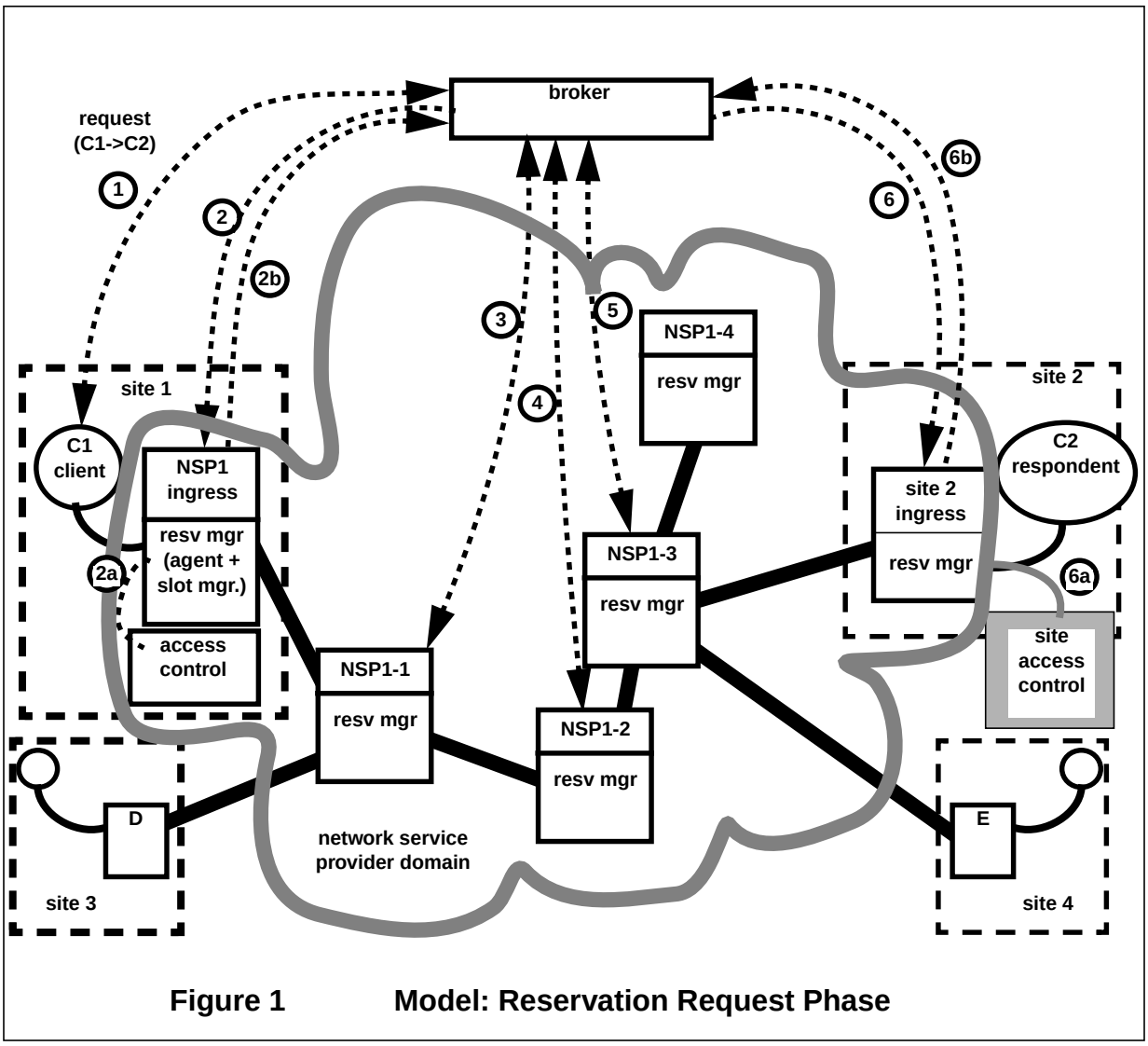


**Figure 1 Model: Reservation Request Phase**

recognize messages that have been digitally signed by them. The broker passes the token representing a a resource manager's successful reservation (as described in Section 1.3) to the next-hop resource manager. The next-hop manager verifies the token's signature and attempts to make the reservation locally. If for some reason the token's signature cannot be validated, the next-hop resource manager refuses to make the reservation and instead returns an error.

The first-hop resource manager, if it can reserve local resources, issues a temporary, or *soft*, reservation token to the broker along with the next-hop information. The broker obtains reservations from the other resource managers in the path as described in Section 1.4. Upon successfully reserving resources down the entire chain of resource managers from sender to receiver, the broker presents all of the reservation tokens it obtained to the first-hop resource manager so that the latter may check the signatures. (We presume that each NSP in the path has a certificate authority, or something like it, that can contacted to provide the relevant public-key information for each resource manager.) If all the tokens are thus verified, the first-hop resource manager changes the soft reservation to a regular, or *hard*, reservation and creates a signed *reservation handle* representing the full reservation path.

The reservation handle is stored in a service authorization server (a secure repository for digitally signed documents) and is assigned a reservation handle identifier, or *reservation ID*. It is this identifier that is returned to the broker for forwarding to the application, and which must be presented to claim the reservation.

A soft reservation eventually expires if not upgraded to a hard reservation. A soft reservation may not be claimed: only a hard reservation can be claimed as described in Section 1.6.

At site 2, the resource manager must request authorization from the site 2 access control system to use this resource (Figure 2, step 6a). For this purpose, C2 must have provided C1 with a *proxy certificate*, that is, a digitally signed document declaring that C1 is authorized to reserve resources on C2's behalf. The exchange of the proxy certificate occurs out of band prior to reservation. C1 provides the proxy certificate to the broker during initial contact (1). The broker presents the proxy when it requests resource reservation from the site 2 resource manager (6). The site 2 resource manage returns its local slot availability but no next-hop information; without a next hop to contact, the broker ends the reservation process.

### 1.6 Bandwidth use (claiming)

Figure 2 illustrates how a reservation is claimed. A client claims a (hard) reservation by pre-

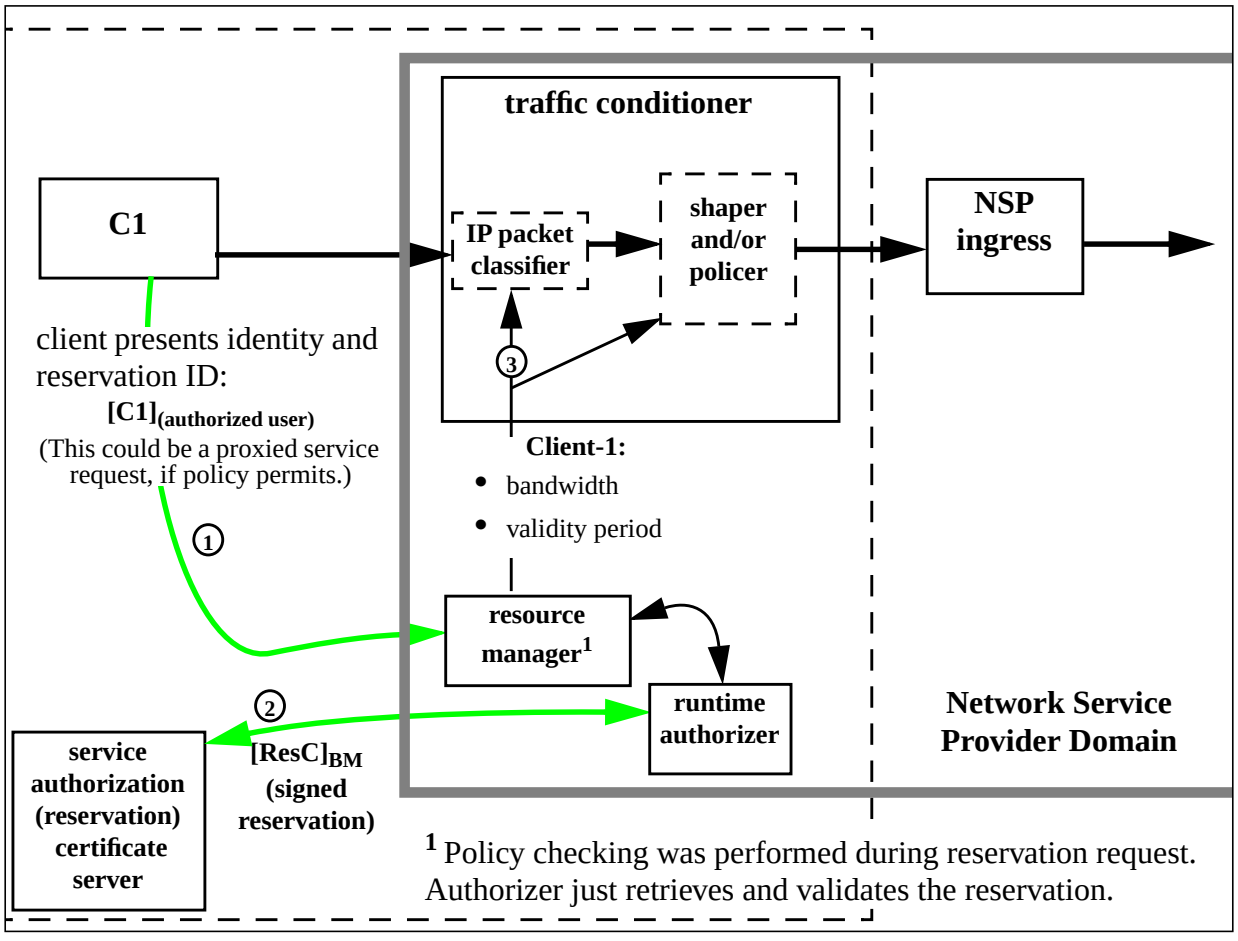


**Figure 2 Model: Claiming Phase**

senting the identity of the claimant (i.e., its own identity, which in the Globus/Akenti environment

will be represented by an X.509 identity certificate) and the reservation ID to the first-hop resource manager (step 1). The resource manager in turn recovers the reservation handle from the service authorization server (2). The reservation handle need only be presented to the first-hop resource manager. The resource managers must trust each other (in fact, the first-hop manager will probably be operated by the NSP), so that per-flow enforcement—that is, traffic conditioning on each application-level flow—can be performed at the first hop.

The first-hop resource manager will have to perform any needed *runtime checks*, that is, any checks that could not be performed during reservation. Such checks could be mandated by stakeholders: a claimant might have to be originating from a particular host or subnet, for example. However, at least one runtime check will probably be required in all cases: the first-hop manager will have to confirm that no topology changes have occurred. If the path has changed and a reservation at a new resource manager is required, the existing full path reservation is considered unfulfillable. We are investigating how best to recover from such a fault.

If, after validating the reservation handle and making any runtime checks, the resource manager finds no problems, it will invoke a network management function that performs the required operations to place the flow into the class specified in the reservation (3). For IP differentiated services, the resource manager would notify the packet classifier and the flow shaper of the flow identity and characteristics. At this point, the flow is tagged and its packets are placed in queues corresponding to the service class of the reservation throughout the network. In the case of IP packets, should any part of the reservation mechanism fail, the packets are merely treated as best-effort traffic.

## 2.0 Related Work

Other proposals for advance reservations in the Internet also implement advance reservation capabilities via cooperating sets of servers that coordinate advance reservations along an end-to-end path [16, 4, 8, 1]. In particular, the NAFUR project [8] addresses many of the same concerns regarding advance reservation of heterogeneous resources, although its emphasis on multimedia, and its implicit assumption of full multicast support throughout the network, indicate a target audience (and possibly service requirements) dissimilar to ours. The work by Olov Schelen and his colleagues on agent-based support for a mix of both advanced and immediate reservation of network bandwidth [4, 10, 11] is closely related to our proposal; in particular, their admission control scheme based on time slots [10] appears to fulfill the requirements for our slot scheduler. None of the foregoing work, however, appears to address the need for end-to-end enforcement of access control policies separate from admission control. The IETF's DiffServ, Policy Framework, RSVP Admission Policy and Integrated Services working groups, among others, are wrestling with the same problems of integrating the low-level quality of service infrastructure under development with high-level admission control and access control mechanisms either proposed or under development [2, 14, 15, 12, 13]. The "policy servers" envisioned in the IETF's COPS proposals, however, are not yet well-defined enough to characterize their relationship to Akenti.

The Darwin project at CMU is building a system with many similarities to the Globus architecture [3]. A resource broker called Xena implements co-allocation strategies and a signaling protocol called Beagle is used to communicate allocation requests to local resource managers that may provide access to network, storage, and compute elements. The concept of hierarchical scheduling

is introduced to allow controlled sharing of network resources managed by different providers: individual providers can specify sharing policies and the Hierarchical Fair Share Curve scheduler is used to determine an efficient schedule that meets all constraints. Like Globus, however, Darwin does not support advance reservations.

Our work builds upon and contributes to the ongoing work to extend the Globus toolkit to support advance reservation of heterogeneous resources [6].

## 3.0 Current Status

We have implemented a prototype slot scheduler and have integrated it into the Globus resource management infrastructure. We are working with Cisco Systems to implement a version of this system on top of its existing RSVP and COPS implementations, and will implement the system on a differentiated services implementation (also to use COPS) when the latter is available.

## 4.0 References

[1] S. Berson and R. Lindell. An architecture for advance reservations in the internet. Technical report. Work in Progress.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. An Architecture for Differentiated Services. Internet Engineering Task Force Request for Comments 2475.

[3] Prashant Chandra, Allan Fisher, Corey Kosak, T. S. Eugene Ng, Peter Steenkiste, Eduardo Takahashi, and Hui Zhang. "Darwin: Resource management for value-added customizable network service." In *Sixth IEEE International Conference on Network Protocols (ICNP'98)*, 1998.

[4] M. Degermark, T. Kohler, S. Pink, and O. Schelen. "Advance Reservations for Predictive Service in the Internet." *ACM/Springer Verlag Journal on Multimedia Systems*, 5(3), 1997.

[5] Ian Foster, Carl Kesselman. "Globus: A Metacomputing Infrastructure Toolkit." *Intl J. Supercomputer Applications*, 11(2):115-128, 1997.

[6] Ian Foster, Carl Kesselman, Craig Lee, Bob Lindell, Klara Nahrstedt, and Alain Roy. "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation." Submitted to *IWQoS '99*.

[7] Ian Foster, Carl Kesselman, Gene Tsudik, Steven Tuecke. "A Security Architecture for Computational Grids." *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pg. 83-92, 1998.

[8] A. Hafid, G. Bochmann, and R. Dssouli. "A quality of service negotiation approach with future reservations (nafur): A detailed study." Technical report, University of Montreal, Montreal, 1996.

[9] Srilekha S. Mudumbai, William Johnston, Mary Thompson, Gary Hoo, and Abdelilah Essiari. "Design and Implementation Issues for A Distributed Access Control System." Submitted to *Fourteenth Annual Computer Security and Applications Conference*.

[10] Olov Schelen, Andreas Nilsson, Joakim Norrgard and Stephen Pink. "Performance of QoS Agents for Provisioning Network Resources." Submitted to *IWQoS '99*.

[11] Olov Schelen and Stephen Pink. "Resource Reservation Agents in the Internet." *Proceedings of the 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'98)*, Cambridge, United Kingdom, July 1998.

[12] S. Shenker and J. Wroclawski. General Characterization Parameters for Integrated Service Network Elements. Internet Engineering Task Force Request for Comments 2215.

[13] S. Shenker and J. Wroclawski. Network Element Service Specification Template. Internet Engineering Task Force Request for Comments 2216.

[14] John Strassner and Ed Ellesson. Policy Framework Core Information Model. Internet Engineering Task Force Internet Draft; work in progress.

[15] Raj Yavatkar, Dimitrios Pendarakis, and Roch Guerin. A Framework for Policy-based Admission Control. Internet Engineering Task Force Internet Draft; work in progress.

[16] L. Zhang, V. Jacobson, and K. Nichols. "A two-bit differentiated services architecture for the internet." Internet Draft, Internet Engineering Task Force, 1997.